Estimating Planetary Habitability via Particle Swarm Optimization of CES Production Functions.

Abhijit J. Theophilus

January 17, 2018

1 Introduction

The search for extra-terrestrial life and potentially habitable extrasolar planets has been an international venture demanding large investments in cost and effort, since Frank Drake's attempt with Project Ozma in the mid-20th century. The first exoplanet was officially confirmed in 1992 which marked the start of a trend that has lasted 25 years and yielded over 3,700 confirmed exoplanets. There have been attempts to assess the habitability of these planets and to assign a score based on their similarity to Earth. Two such habitability scores are the Cobb-Douglas Habitability Score (CDHS) and the Constant Elasticity Earth Similarity Approach (CEESA) score. Estimating these scores involves maximizing a production function while observing a set of constraints on the input variables.

Under most paradigms, maximizing a continuous function requires calculating a gradient. This may not always be feasible for non-polynomial functions in high-dimensional search spaces. Further, subjecting the input variables to constraints, as needed by CDHS and CEESA, are not always straightforward to represent within the model. This paper presents an approach to Constrained Optimization (CO) using the swarm intelligence metaheuristic. Particle Swarm Optimization (PSO) is a method for optimizing a continuous function that does away with the need for a gradient. It employs a large number of particles that traverse the search space converging toward a global best solution encountered by at least one of the particles.

Particle Swarm Optimization is a distributed method that requires simple mathematical operators and short segments of code, making it a lucrative solution where computational resources are at a premium. Its implementation is highly parallelizable. It scales with the dimensionality of the search space. The standard PSO algorithm does not deal with constraints but through variations in initializing and updating particles constraints are straightforward to represent and adhere to, as seen in Section 3.2.

PSO has been adapted to a wide range of design optimization problems including network and VLSI design. It has found applications in machine learning under clustering, feature detection and classification. As a modeling paradigm, it has been used for constructing customer satisfaction models, modeling MIDI music and chaotic time series modeling.

This paper demonstrates the applicability of Particle Swarm Optimization in estimating CDHS and CEESA scores of an exoplanet by maximizing their respective production functions, discussed in Sections 2.1 and 2.2. CDHS considers the planet's Radius, Mass, Escape Velocity and Surface Temperature, while CEESA includes a fifth parameter, the Orbital Eccentricity of the planet. The Exoplanets Catalog hosted by the Planetary Habitability Laboratory, UPR Arecibo records these parameters for each exoplanet in Earth Units. Section **??** reports the performance of PSO and contrasts it against an earlier effort to estimate these scores using Stochastic Gradient Ascent (SGA).

2 Habitability Scores

2.1 Cobb-Douglas Habitability Score

Estimating the Cobb-Douglas Habitability Score (CDHS) requires estimating an interior CDHS ($\mathrm{CDHS_i}$) and a surface CDHS ($\mathrm{CDHS_s}$) by maximizing the following production functions,

$$Y_i \;=\; CDHS_i \;= R^{\alpha}.\, D^{\beta} \tag{1a}$$

$$Y_s \;=\; CDHS_s = {V_e}^{\gamma}.{T_s}^{\delta} \tag{1b}$$

where, R, D, V_e and T_s are density, radius, escape velocity and surface temperature respectively. α, β, γ and δ are the elasticity coefficients subject to,

$$0 < \alpha, \beta, \gamma, \delta < 1 \tag{2}$$

Equations 1a and 1b are convex under either Constant Returns to Scale (CRS) or Decreasing Returns to Scale (DRS) marked by two constraints on the elasticity coefficients,

$$\text{CRS:} \quad \alpha + \beta = 1, \quad \gamma + \delta = 1, \tag{3a}$$

$$\text{DRS:} \quad \alpha + \beta < 1, \quad \gamma + \delta < 1. \tag{3b}$$

The final CDHS is the convex combination of the interior and surface CDHS values as given by,

$$Y \;=\; w_i.Y_i + w_s.Y_s \tag{4}$$

2.2 Constant Elasticity Earth Similarity Approach

The Constant Elasticity Earth Similarity Approach (CEESA) uses the following production function to estimate the habitability score of an exoplanet,

$$Y = (r.R^{\rho} + d.D^{\rho} + t.{T_s}^{\rho} + v.{V_e}^{\rho} + e.E^{\rho})^{\frac{\eta}{\rho}} \tag{5}$$

where, E is the fifth parameter denoting Orbital Eccentricity. The value of ρ lies within $0 < \rho \leq 1$. The coefficients (r, d, t, v and e) are constrained by,

$$0 < r, d, t, v, e < 1 \tag{6a}$$

$$r + d + t + v + e = 1 \tag{6b}$$

The value of η is constrained by the scale of production used,

$$\text{CRS:} \quad 0 < \eta \leq 1, \tag{7a}$$

$$\text{DRS:} \quad \eta = 1. \tag{7b}$$

3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a biologically inspired metaheuristic for finding the global minima of a function. Traditionally designed for unconstrained inputs, it works by iteratively converging a population of randomly initialized solutions, called particles, toward a globally optimal solution. Each particle in the population keeps track of its current position and the best solution it has encountered so far, called *pbest*. Each particle also has an associated randomized velocity used to traverse the search space. The swarm keeps track of the overall best solution, called *gbest*. Each iteration of the swarm updates the velocity of the particle towards its *pbest* and the *gbest* values.

Procedure 1 Algorithm for PSO.

Input: $f(x)$, the function to minimize.
Output: global minimum of $f(x)$.

1: **for** each particle $i \leftarrow 1, n$ **do**
2: $\quad p_i \sim U(l,u)^d$
3: $\quad v_i \sim U(-|u-l|, |u-l|)^d$
4: $\quad pbest_i \leftarrow p_i$
5: **end for**
6: $gbest \leftarrow \underset{pbest_i,\, i=1\ldots n}{\arg\min} f(pbest_i)$
7: **repeat**
8: $\quad oldbest \leftarrow gbest$
9: $\quad$ **for** each particle $i \leftarrow 1 \ldots n$ **do**
10: $\qquad u_p, u_g \sim U(0,1)$
11: $\qquad v_i \leftarrow \mu.v_i + \lambda_g.u_g.(gbest - p_i) + \lambda_p.u_p.(pbest_i - p_i)$
12: $\qquad p_i \leftarrow p_i + v_i$
13: $\qquad$ **if** $f(p_i) < f(pbest_i)$ **then**
14: $\qquad\quad pbest_i \leftarrow p_i$
15: $\qquad$ **end if**
16: $\quad$ **end for**
17: $\quad gbest \leftarrow \underset{pbest_i,\, i=1\ldots n}{\arg\min} f(pbest_i)$
18: **until** $|\, oldbest - gbest \,| < threshold$
19: **return** $f(gbest)$

3.1 PSO for Unconstrained Optimization

Let $f(x)$ be the function to be minimized, where x is a d-dimensional vector. $f(x)$ is also called the fitness function. Algorithm 1 outlines the approach to minimizing $f(x)$ using PSO. A set of particles are randomly initialized with a position and a velocity, where l and u are the lower and upper boundaries of the search space. The position of the particle corresponds to its associated solution. The algorithm initializes each particle's *pbest* to its initial position. The *pbest* position that corresponds to the minimum fitness is selected to be the *gbest* position of the swarm.

On each iteration, the algorithm updates the velocity and position of each particle. For each particle, it picks two random numbers u_g, u_p from a uniform distribution, $U(0,1)$ and updates the particle velocity as indicated in line 11. Here, μ is the friction coefficient and λ_g, λ_p are the global and particle learning rates. If the new position of the particle corresponds to a better fit than its *pbest*, the algorithm updates *pbest* to the new position. Once the algorithm has updated all particles, it updates *gbest* to the new overall best position. A suitable termination criteria for the swarm, under convex optimization, is when the *gbest* position has not changed by the end of the iteration.

3.2 PSO with Leaders for Constrained Optimization

Although PSO has eliminated the need to estimate the gradient of a function, as seen in Section 3.1, it still is not suitable for constrained optimization. The standard PSO algorithm does not ensure that the initial solutions are feasible, and neither does it guarantee that the individual solutions will converge to a feasible global solution. Solving the initialization problem is straightforward, resample each random solution from the uniform distribution until every initial solution is feasible. To solve the convergence problem, each particle uses another particle's *pbest* value, called *lbest*, instead of its own to update its velocity. Algorithm 2 describes this process.

On each iteration, for each particle, the algorithm first picks two random numbers u_g, u_p as before. It then selects a *pbest* value from all particles in the swarm that is closest to the position of the particle being updated as its *lbest*. The *lbest* value substitutes $pbest_i$ in the velocity update equation. While updating *pbest* for the particle, the algorithm checks if the current fit is better than *pbest*, and performs the update

Procedure 2 Algorithm for CO by PSO.

Input: $f(x)$, the function to minimize.
Output: global minimum of $f(x)$.

for each particle $i \leftarrow 1, n$ **do**
  repeat
    $p_i \sim U(l, u)^d$
  until p_i satisfies all constraints
  $v_i \sim U(-|u-l|, |u-l|)^d$
  $pbest_i \leftarrow p_i$
end for
$gbest \leftarrow \underset{pbest_i,\, i=1\ldots n}{\arg\min} f(pbest_i)$
repeat
  $oldbest \leftarrow gbest$
  for each particle $i \leftarrow 1 \ldots n$ **do**
    $u_p, u_g \sim U(0,1)$
    $lbest \leftarrow \underset{pbest_j,\, j=1\ldots n}{\arg\min} \|pbest_j - p_i\|^2$
    $v_i \leftarrow \mu.v_i + \lambda_g.u_g.(gbest - p_i) + \lambda_p.u_p.(lbest - p_i)$
    $p_i \leftarrow p_i + v_i$
    if $f(p_i) < f(pbest_i)$ **and** p_i satisfies all constraints **then**
      $pbest_i \leftarrow p_i$
    end if
  end for
  $gbest \leftarrow \underset{pbest_i,\, i=1\ldots n}{\arg\min} f(pbest_i)$
until $|\, oldbest - gbest \,| < threshold$
return $f(gbest)$

if the current position satisfies all constraints. The algorithm updates *gbest* as before.

4 Representing the Problem

A Constrained Optimization problem can represented as,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_k(x) \leq 0, \qquad k = 1 \ldots q, \\ & h_l(x) = 0, \qquad l = 1 \ldots r. \end{aligned}$$

Introducing an error threshold ϵ can convert strict inequalities of the form ${g_k}'(x) < 0$ to non-strict inequalities of the form $g_k(x) = {g_k}'(x) + \epsilon \leq 0$. Adding a tolerance τ transforms equality constraints to a pair of inequalities,

$$\begin{aligned} g_{(q+l)}(x) &= h_l(x) - \tau \quad && \leq 0, \quad l = 1 \ldots r, \\ g_{(q+r+l)}(x) &= -h_l(x) - \tau \quad && \leq 0, \quad l = 1 \ldots r. \end{aligned}$$

Thus, r equality constraints become $2r$ inequality constraints, raising the total number of constraints, denoted by s, to $s = q + 2r$. For each solution p_i, c_i denotes the constraint vector where, $c_{ik} = \max\{g_k(p_i), 0\},\ k = 1 \ldots s$. When $c_{ik} = 0,\ \forall k = 1 \ldots s$, the solution p_i lies within the feasible region. When $c_{ik} > 0$, the solution p_i violates the k^{th} constraint.

Parameter	Description	Unit
P. Radius	Estimated radius	Earth Units (EU)
P. Density	Density	Earth Units (EU)
P. Esc Vel	Escape velocity	Earth Units (EU)
P. Ts Mean	Surface temperature	Kelvin (K)
P. Eccentricity	Orbital eccentricity	

Table 1: Parameters from the PHL-EC used for the experiment.

Following these guidelines to represent a CO problem, CDHS estimation under CRS becomes,

$$
\begin{aligned}
&\underset{\alpha,\beta,\gamma,\delta}{\text{minimize:}} && Y_i = -R^{\alpha}.D^{\beta},\ Y_s = -{V_e}^{\gamma}.{T_s}^{\delta} \\
&\text{subject to:} && -\phi + \epsilon \le 0, \quad \phi - 1 + \epsilon \le 0, \qquad \forall \phi \in \{\alpha, \beta, \gamma, \delta\} \\
& && (\alpha + \beta - 1) - \tau \le 0, \quad (\gamma + \delta - 1) - \tau \le 0, \\
& && (1 - \alpha - \beta) - \tau \le 0, \quad (1 - \gamma - \delta) - \tau \le 0,
\end{aligned}
\tag{8}
$$

but with DRS the last two constraints for Y_i and Y_s are replaced with,

$$
\begin{aligned}
\alpha + \beta + \epsilon - 1 \le 0, \\
\gamma + \delta + \epsilon - 1 \le 0.
\end{aligned}
\tag{9}
$$

The CEESA score estimation for DRS is represented as,

$$
\begin{aligned}
&\underset{r,d,t,v,e,\rho,\eta}{\text{minimize}} && Y = -(r.R^{\rho} + d.D^{\rho} + t.{T_s}^{\rho} + v.{V_e}^{\rho} + e.E^{\rho})^{\frac{\eta}{\rho}} \\
&\text{subject to} && -\phi + \epsilon \le 0, \quad \phi - 1 + \epsilon \le 0, \qquad \forall \phi \in \{r, d, t, v, e, \eta\} \\
& && \rho - 1 \le 0, \quad \rho - 1 + \epsilon \le 0, \\
& && (r + d + t + v + e - 1) - \tau \le 0, \\
& && (1 - r - d - t - v - e) - \tau \le 0.
\end{aligned}
\tag{10}
$$

Under CRS there is no need for the parameter η. The objective function for the problem becomes,

$$
\underset{r,d,t,v,e,\rho}{\text{minimize}} \quad Y = -(r.R^{\rho} + d.D^{\rho} + t.{T_s}^{\rho} + v.{V_e}^{\rho} + e.E^{\rho})^{\frac{1}{\rho}}. \tag{11}
$$

5 Experiment and Results

The data set used for analysis is the Confirmed Exoplanets Catalog maintained by the Planetary Habitability Laboratory (PHL). The catalog records observed and modeled parameters for exoplanets confirmed by the Extrasolar Planets Encyclopedia. Table 1 describes the parameters from the PHL Exoplanets Catalog (PHL-EC) used for the experiment. Since surface temperature and eccentricity are not recorded in Earth Units, we normalized these values by dividing them with Earth's surface temperature (288K) and eccentricity (0.017). PHL-EC assumes an Eccentricity of 0 when unavailable.

The implemenation uses $n = 25$ particles to traverse the search space with learning rates of $\lambda_g = 0.8$ and $\lambda_p = 0.2$. Velocity is regulated through a friction coefficient of $\mu = 0.6$ and bound by ± 1.0. We have used an error threshold of $\epsilon = 1\text{e}{-6}$ in converting strict inequalities to non-strict inequalities, and a tolerance of $\tau = 1\text{e}{-7}$ when transforming an equality constraint to a pair of inequalities. Further implementation details are discussed in Appendix A.

Tables 2 and 3 record the CDHS values for a sample of exoplanets under CRS and DRS respectively at $w_i = 0.99$ and $w_s = 0.01$. Tables 5 and 4 record the same for the CEESA scores. All tables also record the number of iterations taken to converge to a stable *gbest* value.

Name	Class	α	β	Y_i	i_i	γ	δ	Y_s	i_s	*CDHS*
GJ 176 b	non	0.460	0.540	1.90	50	0.107	0.893	2.11	61	1.90
GJ 667 C b	non	0.423	0.577	1.71	58	0.692	0.308	1.81	54	1.71
GJ 667 C e	psy	0.129	0.871	1.40	50	0.258	0.742	1.39	55	1.40
GJ 667 C f	psy	0.534	0.466	1.40	48	0.865	0.135	1.39	47	1.40
GJ 3634 b	non	0.409	0.591	1.89	58	0.724	0.276	2.09	48	1.89
HD 20794 c	non	0.260	0.740	1.35	50	0.096	0.904	1.34	58	1.35
HD 40307 e	non	0.168	0.832	1.50	49	0.636	0.364	1.53	63	1.50
HD 40307 f	non	0.702	0.298	1.52	68	0.303	0.697	1.55	45	1.52
HD 40307 g	psy	0.964	0.036	1.82	51	0.083	0.917	1.98	55	1.82
Kepler-186 f	hyp	0.338	0.662	1.17	50	0.979	0.021	1.12	40	1.17
Proxima Cen b	psy	0.515	0.484	1.12	37	0.755	0.245	1.07	00	1.12
TRAPPIST-1 b	non	0.319	0.681	1.09	00	0.801	0.199	0.89	00	1.09
TRAPPIST-1 c	non	0.465	0.535	1.06	00	0.935	0.065	1.14	26	1.06
TRAPPIST-1 d	mes	0.635	0.365	0.77	34	0.475	0.525	0.73	47	0.77
TRAPPIST-1 e	psy	0.145	0.855	0.92	00	0.897	0.103	0.83	55	0.92
TRAPPIST-1 g	hyp	0.226	0.774	1.13	43	0.876	0.124	1.09	00	1.13

Table 2: Estimated values of CDHS under CRS by PSO.
α, β, γ and δ record the parameters of Equation 8. Y_i and Y_s record the maxima for the objective functions. i_i and i_s specify the number of iterations taken to converge to a stable *gbest* value. Under the Class column there are four categories for the planets — Psychroplanets (psy), Mesoplanets (mes), Non-Habitable planets (non) and Hypopsychroplanets (hyp). *CDHS* is the final scores with $w_i = 0.99$ and $w_s = 0.01$.

Name	Class	α	β	Y_i	i_i	γ	δ	Y_s	i_s	*CDHS*
GJ 176 b	non	0.395	0.604	1.90	059	0.372	0.627	2.11	056	1.90
GJ 667 C b	non	0.781	0.218	1.71	058	0.902	0.097	1.81	057	1.71
GJ 667 C e	psy	0.179	0.820	1.40	049	0.234	0.765	1.39	060	1.40
GJ 667 C f	psy	0.704	0.295	1.40	064	0.398	0.601	1.39	061	1.40
GJ 3634 b	non	0.602	0.397	1.89	059	0.429	0.570	2.09	077	1.89
HD 20794 c	non	0.014	0.985	1.35	050	0.116	0.883	1.34	045	1.35
HD 40307 e	non	0.752	0.247	1.50	060	0.677	0.322	1.53	050	1.50
HD 40307 f	non	0.887	0.112	1.52	051	0.261	0.738	1.55	060	1.52
HD 40307 g	psy	0.300	0.699	1.82	062	0.785	0.214	1.98	056	1.82
Kepler-186 f	hyp	0.073	0.926	1.17	046	0.740	0.259	1.12	051	1.17
Proxima Cen b	psy	0.045	0.954	1.12	057	0.216	0.783	1.07	053	1.12
TRAPPIST-1 b	non	0.102	0.897	1.09	041	0.000	0.000	1.00	065	1.09
TRAPPIST-1 c	non	0.471	0.528	1.06	044	0.227	0.772	1.14	057	1.06
TRAPPIST-1 d	mes	0.000	0.000	1.00	067	0.000	0.000	1.00	059	1.00
TRAPPIST-1 e	psy	0.000	0.000	1.00	055	0.000	0.000	1.00	057	1.00
TRAPPIST-1 g	hyp	0.888	0.111	1.13	047	0.949	0.050	1.09	046	1.13

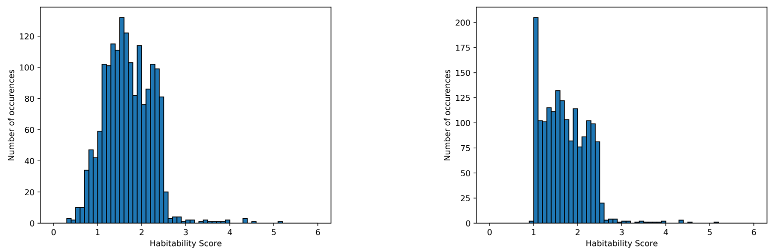
Table 3: Estimated values of CDHS under DRS by PSO.
Here, α, β, γ and δ record the parameters of Equation 9. The other columns are the same as those in Table 2.

Name	Class	r	d	t	v	e	ρ	η	*CEESA*	i
GJ 176 b	non	0.304	0.001	0.375	0.271	0.050	0.467	0.808	1.52	85
GJ 667 C b	non	0.297	0.010	0.318	0.052	0.322	0.682	0.730	2.36	90
GJ 667 C e	psy	0.230	0.286	0.137	0.199	0.148	0.551	0.906	1.14	85
GJ 667 C f	psy	0.397	0.035	0.152	0.402	0.014	0.793	0.999	1.31	100
GJ 3634 b	non	0.178	0.175	0.005	0.194	0.447	0.894	0.657	2.07	94
HD 20794 c	non	0.073	0.142	0.452	0.190	0.144	0.953	0.635	1.20	78
HD 40307 e	non	0.156	0.307	0.185	0.033	0.319	0.428	0.939	2.69	88
HD 40307 f	non	0.272	0.231	0.064	0.305	0.127	0.676	0.802	1.28	77
HD 40307 g	psy	0.113	0.219	0.066	0.454	0.148	0.711	0.991	3.26	92
Kepler-186 f	hyp	0.039	0.159	0.116	0.329	0.357	0.253	0.919	1.35	70
Proxima Cen b	psy	0.272	0.173	0.284	0.193	0.079	0.615	0.114	0.99	75
TRAPPIST-1 b	non	0.488	0.151	0.039	0.193	0.129	0.151	0.014	0.99	87
TRAPPIST-1 c	non	0.172	0.236	0.275	0.242	0.075	0.969	0.962	1.06	80
TRAPPIST-1 d	mes	0.106	0.308	0.075	0.218	0.293	0.844	0.017	0.99	93
TRAPPIST-1 e	psy	0.189	0.266	0.192	0.094	0.260	0.371	0.006	0.99	84
TRAPPIST-1 g	hyp	0.326	0.186	0.143	0.278	0.067	0.315	0.021	1.00	76

Table 4: Estimated values of CEESA under DRS by PSO.
r, d, t, v, e, ρ and eta record the parameters of Equation 10. *CEESA* records the maxima of the objective function. i specifies the number of iterations taken to converge to the maxima.

Name	Class	r	d	t	v	e	ρ	*CEESA*	i
GJ 176 b	non	0.194	0.020	0.315	0.465	0.006	0.398	1.88	86
GJ 667 C b	non	0.162	0.289	0.090	0.087	0.372	0.836	3.54	107
GJ 667 C e	psy	0.373	0.032	0.134	0.304	0.157	0.217	1.25	71
GJ 667 C f	psy	0.394	0.006	0.043	0.360	0.196	0.490	1.44	81
GJ 3634 b	non	0.351	0.122	0.006	0.069	0.453	0.439	2.89	96
HD 20794 c	non	0.101	0.077	0.691	0.071	0.059	0.756	1.58	94
HD 40307 e	non	0.069	0.091	0.097	0.173	0.569	0.768	5.29	94
HD 40307 f	non	0.285	0.161	0.053	0.443	0.058	0.342	1.42	73
HD 40307 g	psy	0.156	0.010	0.081	0.302	0.451	0.612	7.15	94
Kepler-186 f	hyp	0.036	0.017	0.082	0.383	0.483	0.929	1.68	85
Proxima Cen b	psy	0.352	0.383	0.103	0.059	0.103	0.936	0.89	83
TRAPPIST-1 b	non	0.148	0.147	0.344	0.269	0.093	0.767	0.94	81
TRAPPIST-1 c	non	0.038	0.060	0.575	0.321	0.005	0.602	1.17	86
TRAPPIST-1 d	mes	0.023	0.065	0.475	0.391	0.045	0.830	0.84	79
TRAPPIST-1 e	psy	0.176	0.464	0.253	0.103	0.004	0.920	0.86	81
TRAPPIST-1 g	hyp	0.060	0.086	0.310	0.540	0.004	0.848	0.97	86

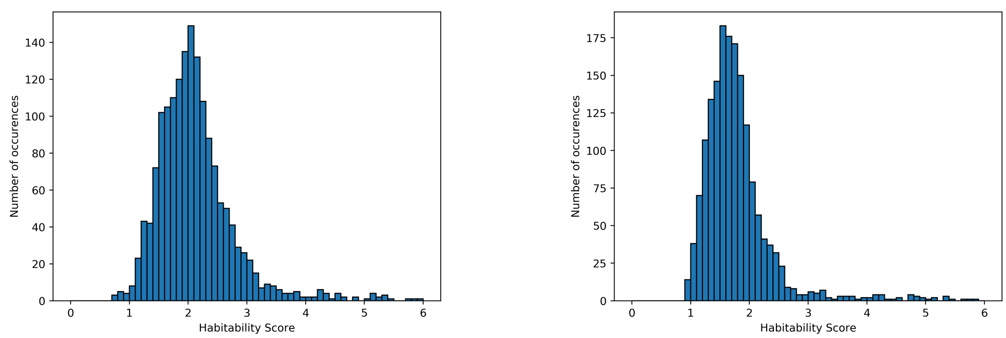
Table 5: Estimated values of CEESA under CRS by PSO.
Here, r, d, t, v, e and ρ record the parameters of Equation 11. Since under the CRS constraint $\eta = 1$, there is no need for the parameter in the problem.



(a) CDHS-CRS

(b) CDHS-DRS

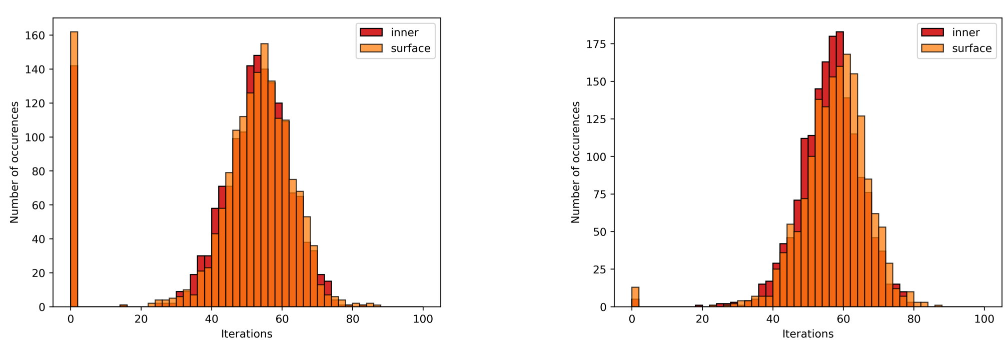
Figure 1: CDHS Distribution Plots



(a) CEESA-CRS

(b) CEESA-DRS

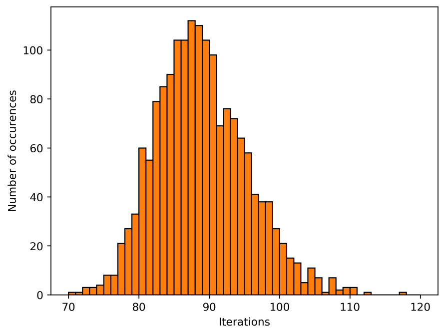
Figure 2: CEESA Distribution Plots



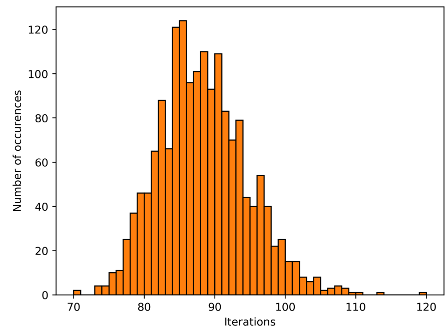
(a) CDHS-CRS

(b) CDHS-DRS

Figure 3: CDHS Iterations Distribution Plots



(a) CEESA-CRS



(b) CEESA-DRS

Figure 4: CEESA Iterations Distribution Plots

6 Conclusions

Should include,

- Why is the speed so important?
- Parallelizable.

A Improving Performance

Matrices P and V represent current position and velocity, where the i^{th} row of each correspond to the position and velocity of particle i. Each row of the matrix L is the leader for the particle in the corresponding row of P. The constraint matrix C is constructed by stacking the constraint vectors c_i described in Section 4. Let r', r'' be two random vectors of length n drawn from the uniform distribution $U(0,1)^n$. Let X_i denote the i^{th} row of matrix X. The implementation of each iteration while updating particles in Algorithm 2 reduces to,

$$
\begin{aligned}
g' &= f([gbest]) \\
L &= [\underset{pbest_j,\ j=1\ldots n}{\arg\min} \|pbest_j - P_i\|^2 \ \mid\ \forall i = 1\ldots n \quad]^T \\
V &= \mu . V + \lambda_g \begin{bmatrix} r_1'(gbest - P_1) \\ r_2'(gbest - P_2) \\ \vdots \\ r_n'(gbest - P_n) \end{bmatrix} + \lambda_p \begin{bmatrix} r_1''(L_1 - P_1) \\ r_2''(L_2 - P_2) \\ \vdots \\ r_n''(L_n - P_n) \end{bmatrix} \\
V &= [\quad \operatorname{sgn}(v_{ij}) * \max\{|v_{max}|, |v_{ij}|\} \ \mid\ \forall i = 1\ldots n, \forall j = 1\ldots d \quad] \\
P &= P + V
\end{aligned}
$$